

**PATENT**  
**5681-58400**  
**P8780**

"EXPRESS MAIL" MAILING LABEL NUMBER  
EV317117001US

DATE OF DEPOSIT JULY 7, 2003

I HEREBY CERTIFY THAT THIS PAPER OR  
FEE IS BEING DEPOSITED WITH THE  
UNITED STATES POSTAL SERVICE  
"EXPRESS MAIL POST OFFICE TO  
ADDRESSEE" SERVICE UNDER 37 C.F.R. §  
1.10 ON THE DATE INDICATED ABOVE AND  
IS ADDRESSED TO THE COMMISSIONER OF  
PATENTS, MS PATENT APPLICATION, P.O.  
BOX 1450, ALEXANDRIA, VA 22313-1450.



Derrick Brown

## System and Method for Dynamically Disabling Partially Streamed Content

By:

Sridhar Satuloori  
Sivasankaran R

Robert C. Kowert/CLJ  
Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.  
P.O. Box 398  
Austin, TX 78767-0398  
Ph: (512) 853-8800

## **BACKGROUND OF THE INVENTION**

### **Field of the Invention**

5    **[0001]**     This invention relates to the field of network computer systems and, more particularly, to a mechanism for disabling content partially streamed to a client.

### **Description of the Related Art**

10   **[0002]**     The client/server relationship is widespread in modern networked computing, particularly in the domain of the Internet and World Wide Web. For example, a client may contact a server via a network and request a service or a piece of information. A web server may provide copies of various web pages to a plurality of clients. More specifically, a client may contact the web server with the URL of a specific web page,  
15   which may then be sent to the client and displayed for the user.

**[0003]**     As described above, web servers primarily retrieve files from one or more data sources and transmit the information therein over a network. Alternatively or additionally, servers may execute significantly more computation in response to a client  
20   request. For example, a website may use a database in tandem with common gateway interface (CGI) scripts to provide a more dynamic content retrieval system. In such an example, a client may request information related to a certain topic from the server. The server may then execute a search for the topic on an internal database, and dynamically assemble a web page with the desired content before returning the results to the client.  
25   Additional computation may be executed on the server when handling other web-based applications, such as electronic commerce.

**[0004]**     Because such extra computation or large content sizes may increase the turnaround time for the server responding to a client's request, the server may partially  
30   stream content, including hyperlinks, to a client while continuing to process the request.

As additional content is assembled, the server may continue streaming content until the request has been fully satisfied.

[0005] However, after such content is sent to the client, it cannot typically be recalled.

- 5 This may pose a problem if the server encounters any difficulties in its continuing calculations or request processing after sending partial content to the client. For example, the server may detect a security violation in further processing that makes it necessary to deny the client access to hyperlinks that have already been sent. Alternatively, the server may receive an error message or time out from a resource necessary to complete the client
- 10 request, such as a database. In this scenario, the entire session may rendered invalid.

## **SUMMARY**

[0006] A system and method for dynamically disabling partially streamed content may include a server receiving a request from a client. A session may be initiated or  
5 continued in response to the request. The server may begin to stream content to the client as a partial fulfillment of the request. While processing the request, the server may determine if the partially streamed content should be disabled. In response to determining that the partially streamed content should be disabled, the server may disable the partially  
10 streamed content without terminating the session. Disabling the partially streamed content may involve preventing the client from accessing content referenced by a hyperlink associated with the partially streamed content, or may involve the use of a controller located on the client to disable content streamed to the client.

[0007] In one embodiment, a method for dynamically disabling partially streamed  
15 content may include a server processing a request from a client. The request may initiate or continue a session between the server and the client. The method may further include streaming out content from the server to the client as a partial fulfillment of the request. While processing the request, the method may include determining if the content should be disabled, and if the content should be disabled, disabling the content without  
20 terminating the session. The content may be disabled by invalidating links associated with the partially streamed content and/or by instructing a controller on the client to disable partially streamed content received by the client.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

[0008] Fig. 1 is a flow diagram illustrating a method for dynamically disabling partially streamed content, according to one embodiment.

5

[0009] Fig. 2 is a block diagram of one embodiment of a system for dynamically disabling partially streamed content.

[0010] Fig. 3 is a flow diagram illustrating an embodiment of a method for dynamically disabling partially streamed content including invalidating hyperlinks.

10

[0011] Fig. 4 is a block diagram of one embodiment of a system for dynamically disabling partially streamed content, including one or more controllers.

[0012] Fig. 5 is a flow diagram illustrating an embodiment of a method for dynamically disabling partially streamed content including using a controller.

15

[0013] Figs. 6A and 6B are flow diagrams illustrating embodiments of a method for disabling partially streamed content including using a controller.

20

[0014] Fig. 7 illustrates an exemplary server for implementing certain embodiments.

[0015] While the invention is susceptible to various modifications and alternative forms, specific embodiments are shown by way of example in the drawings and are herein described in detail. It should be understood, however, that drawings and detailed description thereto are not intended to limit the invention to the particular form disclosed, but on the contrary, the invention is to cover all modifications, equivalents and alternatives falling within the spirit and scope of the present invention as defined by the appended claims.

25

30

## **DETAILED DESCRIPTION OF EMBODIMENTS**

[0016] Turning now to Fig. 1, a flow diagram illustrating a method for dynamically disabling partially streamed content is shown. As will be described below, the method may be executable on a server to disable partially streamed content to a client. In one embodiment, content may include one or more hyperlinks which lead back to additional information on the server, for example. In another embodiment, content may include one more pieces of information calculated or provided as a result of a request from a client to an application on the server, as will be described below.

[0017] In 100, the server receives a request from a client. The request may begin the execution of an application process on the server, thereby initiating a session between the client and the application on the server. Or the request may be made as part of an existing session. In one embodiment, a session may comprise multiple requests and responses between the application and the client, wherein each request is operable to further control, modify, or inform the application, and each response further informs the client of the results of the request.

[0018] In 102, the application, operating through the server, may begin to stream content to the client as a partial fulfillment of the request from the client, while continuing to process the request. Content may be partially streamed out to minimize latency in the interaction between client and application. For example, in one embodiment, a request may take a significant amount of time to process entirely. The application may therefore begin to stream some results to the client as the results are determined, rather than waiting for all results to be determined before streaming all results as a single block to the client.

[0019] In 104, the client may receive content and begin to display or act upon the results therein. In one embodiment, the client may allow an end user to access one or more hyperlinks contained within the content. Accessing the hyperlinks may in turn

initiate a new request to the server within the context of the session, thereby allowing the client access to additional content located on or controlled by the server.

5 [0020] In 106, the server may determine that some or all of the streamed content may be invalid. In one embodiment, this determination may derive from a security violation within the application, wherein the application or server determines that the client or some other process involved in the request should not have been allowed access to one or more pieces of content, for example. In another embodiment, this determination may derive from a timeout while handling the request, wherein the application or server  
10 determines that sufficient time has passed during the processing of the request to render some or all of the streamed content invalid. Other reasons are also contemplated for determining that some or all of the partially streamed content is or may be invalid.

[0021] If, in 106, the server determines that no streamed content is invalid, the server  
15 continues to process the request in 108. Alternatively, if the server determines that some of the streamed content is invalid, the server may determine which client or clients and which content should be disabled, as indicated by 110. In one embodiment, the server may examine the session data to determine which clients have received invalid content. The server may then take one or more steps to disable the content determined to be  
20 invalid. For example, the server may prevent the client that received invalid content from accessing one or more hyperlinks associated with the content, as will be described below. In another embodiment, the server may communicate with a controller on the client to disable the invalid content streamed to the client. It is noted that, in disabling the content, the server may not terminate the session between the client and the application, thereby  
25 allowing the client to continue utilizing the application and request other content without having to restart the session.

[0022] Fig. 2 is a block diagram of one embodiment of a system for dynamically disabling partially streamed content. Server 200 may communicate with any of the  
30 clients 230A – C via network 220. Clients 230A – C may be operable to send requests to

one or more applications 210A – C on server 200, and receive content back, as described above. Network 220 may be a point-to-point fabric, a local area network (LAN), a wide area network (WAN), the Internet, any other type of computing interconnect, or a combination thereof. For the purposes of clarity it is noted that any notation in the form of, for example, A – C, may refer to any or all of the elements in the group A – C.

[0023] Applications 210A – C may be any type of application executable in a networked system. Such applications may include, but are not limited to, database applications, web applications, file server applications, or e-commerce applications, for example. Server 20 may also include a content disabling mechanism 212, as will be described below.

[0024] Content disabling mechanism 212 may implement logic to disable access to content that has been streamed out to one or more clients. In one embodiment, content disabling mechanism 212 may exist as one or more utilities or services provided in server 200 which provides various functionality to applications 210A – C. This content disabling functionality may include, but not be limited to, functionality for tracking streamed content and taking steps to disable content determined to be invalid, such as disabling hyperlinks and/or relaying content disabling messages to one or more client controllers. In one embodiment, content disabling mechanism 212 may indicate that a given hyperlink is invalid by associating an error message with the hyperlink, as will be described below.

[0025] In one embodiment, content disabling mechanism 212 may include or be interoperable with a session tracking system operable to monitor a plurality of sessions running between applications 210A – C and a plurality of end users on a plurality of clients. Such a session tracking system may additionally monitor information including, but not limited to, session ID, client ID, user ID, client address, streamed content per session, requests pending and/or requests serviced.

30



[0026] In one embodiment, content disabling mechanism 212 may include or be interoperable with a content lookup table, which may be operable to provide a lookup service connecting one or more streamed hyperlinks to one or more pieces of content or web pages served by server 200. For example, in one embodiment, a client 230A – C  
5 may send a request for content associated with a hyperlink to an application 210A – C on server 200. In response, application 210A – C may use a content lookup table to determine which content is associated with the hyperlink. Application 210A – C may then return the content associated with the hyperlink to client 230A – C.

10 [0027] Turning now to Fig. 3, a flow diagram illustrating a more specific embodiment of the method illustrated in Fig. 1 is shown. As will be described below, the method may be executable on a server to dynamically disable partially streamed content to a client.

[0028] Referring collectively now to Figs. 2 – 3, in 300, server 200 receives a request  
15 from a client 230A – C. The request may begin a session or be part of an existing session between the client and an application 210A – C on the server. As described in Fig. 1, in one embodiment, the session may comprise multiple requests and responses between application 210 A – C and client 230A – C.

20 [0029] In 302, application 210A – C operating through server 200 may begin to stream content to client 230A – C as a partial fulfillment of the request from client 230A – C, while continuing to process the request. Content may be partially streamed out to minimize latency in the interaction between client and application, as described in Fig. 1. It is noted that in one embodiment, content disabling mechanism 212 may track specific  
25 information on which links streamed to which client lead to which content on server 200.

[0030] In 304, client 230A – C may receive content and begin to display or act upon the results therein. In 306, content disabling mechanism 212 may determine that some or all of the content which has been streamed to client 230A – C may be invalid, as  
30 described in Fig. 1.

[0031] If, in 306, content disabling mechanism 212 determines that no streamed content is invalid, application 230A – C may continue to process the request in 308. Alternatively, if content disabling mechanism 212 determines that the streamed content is invalid, application 230A – C may advance to 310, wherein content disabling mechanism 212 determines which content for which client(s) should be disabled, as described in Fig. 1. In one embodiment, disabling content may include disabling one or more links that have been sent to a client within a session. Content disabling mechanism 212 may cause server 200 to return an error message when any of the disabled hyperlinks are accessed by client 230A – C. In one embodiment, to disable hyperlinks content disabling mechanism 212 may invalidate one or more entries in a hyperlink look-up table that corresponds hyperlinks sent to a client within a particular session to particular content.

[0032] Fig. 4 is a block diagram of one embodiment of a system for dynamically disabling partially streamed content, similar to Fig. 2. However, in addition to content disabling mechanism 212, server 200 may also utilize one or more controllers 400 in various embodiments to disable partially streamed content, as will be described below.

[0033] The functionalities of server 200, application 210A – C, network 220, and clients 230A – C are broadly similar to the functionalities described above in Fig. 2. In addition to the functionality described above in Fig. 2, content disabling mechanism 212 may be able to remotely control controllers 400, as described below.

[0034] A Controller 400 may be operable on clients 230A – C to disable content that has been streamed to clients 230A – C from server 200. In one embodiment, a controller 400 may prevent a client 230A – C from accessing hyperlinks which reference content stored on server 200, as directed by content disabling mechanism 212. For example, controller 400 may interact with a client browser application to deactivate certain links or block certain links. In one embodiment, controller 400 may be operable to prevent clients 230A – C from accessing hyperlinks by disabling those hyperlinks on a browser display,

while in another embodiment, controller 400 may be operable to return an error message whenever a disabled hyperlink is accessed.

[0035] Controller 400 may additionally or alternatively be operable to disable content on a client in other ways. For example, controller 400 may prevent content from being copied or saved to another location. Other ways of disabling content on a client are contemplated. In one embodiment, controller 400 may cache content and prevent any such content from being displayed until content disabling mechanism 212 sends an indication that the content is verified as valid, for example.

[0036] In one embodiment, controller 400 may already be installed on a client 230A – C prior to the initiation of a session, as illustrated by controller 400I at client 230B in Fig. 4. For example, a controller 400 may be a separate process running on the client, or may be built in to a client browser application. Alternatively, a controller 400 may be sent to a client 230A – C by server 200, as illustrated by client 230C in Fig. 4, for example. As will be described below, in one embodiment controller 400 may be sent to the client at the beginning of a session, while in another embodiment, controller 400 may be sent to the client after content disabling mechanism 212 has determined that certain content on a given client should be invalidated.

[0037] In one embodiment, a controller may be specific to a given application 210A – C. For example, controller 400A on client 230A in Fig. 4 may be provided by or specified by application 210A. In one embodiment, each application 210A – C may directly control an associated controller 400. In another embodiment, content disabling mechanism 212 to communicate with and control the associated controller 400. In one embodiment, an interface to a controller 400 may be specified in a descriptor file or configuration file on server 200. Content disabling mechanism 212 may access the descriptor or configuration file to locate an appropriate controller for an application and determine how to communicate with the controller and what functions are supported by the controller. In some embodiments, all controllers may follow a standard interface.

[0038] In one embodiment, server 200 may provide a controller 400 as a default controller (represent by controller 400S in Fig. 4) that may be used for all applications, or for applications that do not specify their own controller.

5

[0039] Controllers 400 may be installed on clients 230A – C prior to the beginning of a session, at the beginning of a session, or upon the determination that streamed content is invalid.

10 [0040] Turning now to Fig. 5, a flow diagram illustrating a more specific embodiment of the method illustrated in Fig. 1 is shown. As will be described below, the method may be executable on a server to dynamically disable partially streamed content to a client by instructing a controller to prevent disabled content from being accessed or used, for example.

15

[0041] Referring collectively now to Figs. 4 – 5, in 500, server 200 receives a request from a client 230A – C. The request may begin the execution of an application 210A – C on server 200, thereby initiating or continuing a session between client 210A – C and application 210A – C on the server. As described in Fig. 1, in one embodiment, the session may comprise multiple requests and responses between application 210 A – C and client 230A – C.

25 [0042] In 502, application 210A – C operating through server 200 may begin to stream content to client 230A – C as a partial fulfillment of the request from client 230A – C, while continuing to process the request. Content may be partially streamed out to minimize latency in the interaction between client and application, as described in Fig. 1. Content disabling mechanism 212 may track specific information on which links streamed to which client lead to which content on server 200, as previously described in Fig. 3.

30

[0043] In 504, client 230A – C may receive content and begin to display or act upon the results therein. In 506, content disabling mechanism 212 may determine that some or all of the content which has been streamed to client 230A – C may be invalid, as described in Fig. 1.

5

[0044] If, in 506, content disabling mechanism 212 determines that no streamed content is invalid, application 230A – C may continue to process the request in 508. Alternatively, if content disabling mechanism 212 determines that the streamed content is invalid, application 210 A – C may advance to 510, wherein content disabling mechanism  
10 212 determines which content for which client(s) should be disabled, as described in Fig. 1. Content disabling mechanism 212 may then operate controller 400 on client 230A – C, to prevent client 230A – C from using or from accessing the content to be disabled.

[0045] In some embodiments, in disabling the content, the server may allow the  
15 session to continue between the client and the application, thereby allowing the client to continue utilizing the application and retrieve content without restarting the entire session. Thus, certain content may be disabled while still preserving the state of the session and/or prior results.

[0046] Figs. 6A and 6B both show a flow diagram illustrating a specific embodiment  
20 of the method illustrated in Fig. 5. In both illustrations, the method and functionalities described are substantially similar. However, Fig. 6A illustrates a controller being sent (or downloaded) to a client 230A – C after the initiation of a session, whereas Fig. 6B illustrates a controller being sent after the determination that partially streamed content is  
25 invalid.

[0047] Turning now to Fig. 6A and referring collectively to Figs. 4 – 6A, in 600, server 200 receives a request from a client 230A – C. The request may initiate a session between client 210A – C and application 210A – C on the server. As described in Fig. 1,

in one embodiment, the session may comprise multiple requests and responses between application 210 A – C and client 230A – C.

5 [0048] In 602, server 200 may send a controller 400 to a client 230C. In one embodiment, the server may download the controller to the client as part of the session initiation. In one embodiment, the server may determine whether or not an appropriate controller is already installed on the client, and download the controller only if an appropriate controller is not already installed on the client. In one embodiment, the controller may comprise one or more Java classes, while in another embodiment, the  
10 controller may be an executable or an interpreted script, for example. In one embodiment, the controller may be associated with a web browser, while in another embodiment, the controller may be as stand-alone program or utility.

[0049] In 604, application 210A – C operating through server 200 may begin to  
15 stream content to client 230A – C as a partial fulfillment of the request from client 230A – C, while continuing to process the request. Content may be partially streamed out to minimize latency in the interaction between client and application, as described in Fig. 1. The server, or content disabling mechanism 212, may then track specific information on which links streamed to which client lead to which content on server 200.

20

[0050] In 606, client 230A – C may receive content and begin to display or act upon the results therein. In 608, the server, or content disabling mechanism 212, may determine that some or all of the content which has been streamed to client 230A – C may be invalid, as described in Fig. 1.

25

[0051] If, in 608, the server, or content disabling mechanism 212, determines that no streamed content is invalid, application 230A – C may continue to process the request in 610. Alternatively, if content disabling mechanism 212 determines that the streamed content is invalid, content disabling mechanism 212 may determine which client and  
30 which content should be disabled, as indicated at 612. The server, or content disabling

mechanism, may invalidate content streamed to a client by instructing the controller on the client, as indicated at 614.

[0052] Turning now to Fig. 6B and referring collectively to Figs. 4 – 6B, in 650, server 200 receives a request from a client 230A – C. The request may initiate or continue a session between client 210A – C and application 210A – C on the server. As described in Fig. 1, in one embodiment, the session may comprise multiple requests and responses between application 210 A – C and client 230A – C.

[0053] In 652, application 210A – C operating through server 200 may begin to stream content to client 230A – C as a partial fulfillment of the request from client 230A – C, while continuing to process the request. Content may be partially streamed out to minimize latency in the interaction between client and application, as described in Fig. 1. Content disabling mechanism 212 may then track specific information on which links streamed to which client lead to which content on server 200.

[0054] In 654, client 230A – C may receive content and begin to display or act upon the results therein. In 656, server 200 may determine that some or all of the content which has been streamed to client 230A – C may be invalid, as described in Fig. 1.

[0055] If, in 656, server 200, or content disabling mechanism 212, determines that no streamed content is invalid, application 230A – C may continue to process the request in 658. Alternatively, if it is determined that the streamed content is invalid, the server, or content disabling mechanism 212, may determine which content for which client(s) should be disabled, as described in Fig. 1.

[0056] In 662, server 200 may send a controller 400 to a client 230C upon determining that content needs to be disabled for that client. In one embodiment, the controller may comprise one or more Java classes, while in another embodiment, the controller may be an executable or an interpreted script, for example. In one

embodiment, the controller may be associated with a web browser, while in another embodiment, the controller may be as stand-alone program or utility. In 664, the server, or content disabling mechanism 212, may operate controller 400 to prevent client 230A – C from accessing any content. For example, the controller may prevent copying or saving  
5 of the content. Or the controller may prevent further display of the content. Or the controller may disable hyperlinks in the content. In some embodiments, the controller may interact with a client browser application to perform these various types content disabling.

10 **[0057]** It is noted that various embodiments may combine the various aspects of the functionalities discussed herein. For example, content disabling mechanism 212 may directly manage hyperlinks, while utilizing controller 400 to prevent a client from saving, copying or printing partially streamed content.

15 **[0058]** Turning now to Fig. 7, an exemplary computer system 700 is shown. Computer system 700 may include main memory 720, which may be coupled to multiple processors 710A – B, and I/O interface 730. It is noted that the number of processors is purely illustrative, and that one or more processors may be resident on the system. I/O interface 730 may connect to network interface 760, hard disk drive 740, display adapter  
20 750, and/or removable storage adapter 770. Such a system is exemplary of a laptop, desktop, server, workstation, or other computing device.

**[0059]** Processors 710A – B may be representative of any of various types of processors such as an x86 processor, a PowerPC processor or a CPU from the SPARC  
25 family of RISC processors. Likewise, main memory 720 may be representative of any of various types of memory, including DRAM, SRAM, EDO RAM, SDRAM, DDR SDRAM, Rambus RAM, etc., or a non-volatile memory such as a magnetic media, e.g., a hard drive, or optical storage. It is noted that in other embodiments, main memory 720 may include other types of suitable memory as well, or combinations of the memories  
30 mentioned above.



[0060] As described in detail above in conjunction with Figs. 1 – 6B, processors 710A – B of computer system 700 may execute software configured to execute a method for dynamically disabling partially streamed content. The software may be stored in  
5 memory 720 of computer subsystem 700 in the form of instructions and/or data that implement the operations described above.

[0061] For example, Fig. 7 illustrates an exemplary server 200 and content disabling mechanism 212 stored in main memory 720. The instructions and/or data that comprise  
10 these and all other included components may be executed on one or more of processors 710A – B, thereby implementing the various functionalities of the method and system described in Figs. 1 – 6B.

[0062] Various embodiments may further include receiving, sending or storing  
15 instructions and/or data that implement the operations described above in conjunction with Figs. 1-6B upon a computer readable medium. Generally speaking, a computer readable medium may include storage media or memory media such as magnetic or optical media, e.g. disk or CD-ROM, volatile or non-volatile media such as RAM (e.g. SDRAM, DDR SDRAM, RDRAM, SRAM, etc.), ROM, etc. as well as transmission  
20 media or signals such as electrical, electromagnetic, or digital signals conveyed via a communication medium such as network and/or a wireless link.

[0063] Although the embodiments above have been described in considerable detail, numerous variations and modifications will become apparent to those skilled in the art  
25 once the above disclosure is fully appreciated. It is intended that the following claims be interpreted to embrace all such variations and modifications.